Sub-Grid based Key Vector Assignment: A Key Pre-Distribution Scheme For Distributed Sensor Networks

R. KALINDI, R. KANNAN, S. S. IYENGAR and A. DURRESI
Department of Computer Science, Louisiana State University, LA, USA
Email: {rkalid1,rkannan,iyengar,durresi}@bit.csc.lsu.edu

Received: November 5 2005; revised: January 18 2006

***Abstract*— Security in sensor networks is more important than traditional networks as they are deployed in hostile environments and are more prone to capture. Trusted third party authentication schemes, public-key systems are not suitable owing to their high resource requirements. Key pre-distribution was introduced in [3] to solve this problem. Our scheme achieves identical connectivity compared to the random key pre distribution [4] using a less number of preloaded keys in each sensor node. The design of our scheme is motivated by the observation that at present most key pre-distribution schemes employ random mechanisms which use a large number of keys and are unsuitable for sensor networks.**

In this paper we extend the deterministic key pre-distribution scheme proposed by us in our earlier work [1], which is based on assigning keys to sensors by placing them on a grid. This approach has been further modified to use multiple mappings of keys to nodes. In each mapping every node gets distinct set of keys which it shares with different nodes. The key assignment is done such that, there will be keys in common between nodes in different sub-grids. After randomly being deployed, the nodes discover common keys, authenticate and communicate securely. The analysis and simulation results show that this scheme is able to achieve better security compared to the random schemes.

***Index Terms*— Sensors, grid, pre distribution**

I. INTRODUCTION

Distributed Wireless Sensor Networks (DSN) consists of numerous tiny sensors deployed at high density in regions requiring surveillance and monitoring. These sensors are memory as well as energy constrained. A typical sensor node consists of one or more sensing elements (motion, temperature, pressure, etc.), battery, low power radio transmitter/receiver, microprocessor and limited memory. An important aspect of such networks is that the nodes are unattended , have un-replenishable energy and network topology is unknown. Sensor networks deployed in a hostile environment are prone to different types of malicious attacks like eavesdropping, masquerading, traffic-analysis, etc. To provide security communication should be encrypted and authenticated. Traditional secure communication schemes [8], [9] are not suited for sensor networks as they are more demanding in memory and computationally intensive.

This work was supported by NSF grants, IIS-0312632 and IIS-0329738 and DARPA and AFRL grant # F30602-02-1-0198.

The fundamental constraints under which the sensor networks operate prohibit them from using public key cryptosystems, third party authentication systems etc. These constraints are.

- *Resource Constraints*: A typical tiny sensor node has about 20-30 joules of initial energy. This imposes a strong restriction on the processing capabilities and available memory in a sensor node. For example, a Berkeley mote has a 8-bit 4 MHZ processor which supports a minimal RISC-like instruction set without support for multiplication and other complex operations. Perrig *et al.* [4] showed that a simple RSA operation takes the order of tens of seconds on this processor. Moreover it has less than 4KB of memory after the node is loaded with the necessary operating system and applications. Hence the number of keys and the algorithms that can be stored on a node is very less.
- *Short Radio Range*: Sensor nodes are equipped with low power radio transmitters which have a very small transmission range of less than 20 meters. Thus a DSN uses multi-hop routing for sensor node to base station communication. Henceforth a base station cannot manage the key distribution for the nodes as it incurs high overhead.
- *Hostile Environments*: DSN's are deployed in hostile environments which make the nodes prone to capture. Using key distribution servers to establish shared keys is not feasible because a key distribution server if captured can disclose a large number of keys and thus is a single point of failure.

Identifying these limitations our work extends the seminal work done in this area by Eschenauer *et al.* [3]. They introduced a random key pre distribution scheme in which each node is loaded with a set of keys randomly selected from a key pool. After the deployment phase neighboring nodes exchange information to establish common shared keys which are later used for secure communication. The basic idea behind

this scheme is to have a large pool of keys, from which a set of keys is randomly chosen and stored in each of the sensor nodes. Any two nodes which are able to find common keys within their key subsets can use those shared keys for secure communication and authentication. The key idea of this method is relegate the key establishment process from a key distribution server to the individual nodes thereby making it viable in hostile environments.

Key pre-distribution schemes [3], [5], [4] where key information is distributed to all sensor nodes prior to deployment is the most feasible method for secure communication between resource-constrained devices. A naive way of achieving complete connectivity for a network of N nodes is to have N-1 keys stored in each sensor node. But this method is infeasible due to memory constraints on sensor nodes. To tackle the memory constraints problem a single key can be used network-wide for encrypting data. Although this approach has the least storage cost it is most vulnerable to attack as compromising a single node will cause the entire network's security to be breached.

Chan *et al.* [4] have extended the basic random scheme to enhance the security and resilience of the network using q-compositeness and multi-path key reinforcement. In the q-composite scheme instead of nodes sharing single key they are required to share at least q keys to establish communication. This method claims to achieve higher security under the assumption that network is more prone to small scale attacks and is unlikely to be subject to a large-scale attack. However a higher value of q makes the network less scalable and connectivity is reduced. In the multi-path key reinforcement scheme security is strengthened between any two nodes by exchanging information between the two nodes using multiple paths. In this method although an increase in the number of disjoint paths increases the security, communication overhead increases substantially.

The drawback of the above proposed schemes is that they are not suitable for large scale sensor networks as they require each node to be loaded with a large number of keys. Perrig, in his work in 2001 [5], showed that it is not feasible to implement public key cryptographic protocols in sensor nodes. These sensor nodes have less than 4KB of free memory after loading the necessary Operating System and other necessary applications. Implementation of key distribution schemes presented in [3] results in a requirement of memory for around 200 keys, thus occupying more than half the available memory. This aspect makes the previous proposed schemes impractical for large networks.

The motivation for our scheme is to reduce the number of keys to be loaded in each node. We propose a novel scheme, in which keys are assigned in a deterministic fashion so that any random deployment yields very high connectivity as well as high security among sensor nodes. The scheme is so modeled so as to maximize the connectivity with a small number of keys loaded in each of the sensor nodes. Our scheme requires as few as 25 keys to be stored by each node thus minimizing required memory space in sensors and this enables implementation of our scheme more practical in these sensor nodes.

Any two nodes that share keys have atleast two keys in common. Thus our method is inherently 2-composite. For any two nodes to securely establish a common shared key they need to have exactly two common keys. This restriction makes the scheme more secure against node captures as described in the key discovery phase.

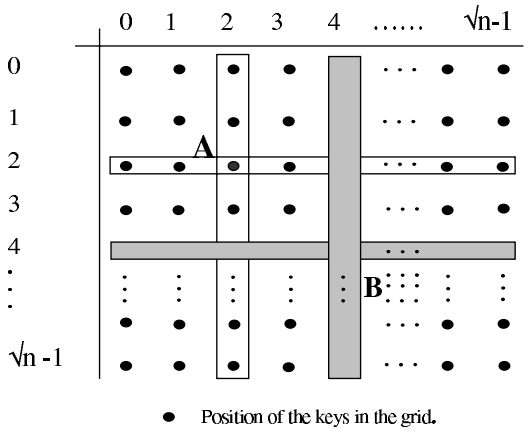


Fig. 1. Simple Grid Scheme key vector assignment

Our Contributions are summarized as below

- First we propose a deterministic key pre-distribution scheme in which every node shares at least two keys with every other node. Each node has $2 \times \sqrt{N} - 1$ keys, where N is the number of nodes in the network. This scheme although guarantees complete connectivity is infeasible owing to its large number of keys. Then we extend this simple scheme to our novel sub-key vector assignment scheme. This scheme trades off connectivity for key space thereby making the network more scalable.
- Then we extend this simple scheme such that each node get a set of different keys from M mappings. In each mapping every node gets distinct set of keys. The final key set of the node is the combined set of keys from all the mappings.
- We use various metrics to show the tradeoffs in connectivity and memory and compare our proposed protocol with the random key pre-distribution scheme.
- We present analytical and simulation results to study the performance of our scheme in terms of security and connectivity.

Most of the previously proposed schemes tackle the issue of connectivity alone while assigning keys. In our proposed schemes we evaluate both connectivity and security. To analyze the performance of our schemes we use three metrics:

- Probability P that two nodes share a common key is a good indicator of the connectivity of the network. We use this metric to show the effectiveness of the proposed protocols.
- The variance V in the number of keys revealed per node, when x nodes are captured, which indicates the security performance of the scheme.
- $\frac{P}{V}$ is the new metric proposed by us which combines the connectivity and security performance of the protocol.

A. Impersonation

This is an important factor that must be considered in thye deisgn of the algorithm. When node is captured all the keys for that particlular node are known to the adversary. Also, a significant number of keys are shared by other nodes. Hence an adversary upon capturing a node can masquerade as a node and try to estbalish shared keys with other nodes. Shared keys among nodes should not reveal any information about the pattern in which nodes shares keys. Our proposed algorithm does that exactly to minimize impersonation. Our scheme is resilient to impersonation because more number of nodes have to be captured to disclose all the keys of uncaptured nodes.

The rest of the paper introduces the basic subgrid scheme and some results on security are provided. Then we introduce the multiple-mapping scheme and evaluate its performance.

II. Proposed Key Pre-Distribution Scheme

This section describes the steps required to establish pair-wise secret keys among sensor nodes. The following three steps are essential in a key pre-distribution scheme.

- Key pre-distribution phase in which every node is loaded with a set of keys V_i(key vector), which are generated by the key server.
- Key discovery phase where every pair of neighboring nodes N_i, N_j finds a key path. The key path is a direct link if $|V_i \bigcap V_j| = 2$ (N_i, N_j share exactly two keys). If neighboring nodes do not share exactly two keys they find a logical path $\mathcal{P}_{ij}$ through a set of intermediate nodes N_1, N_2.... N_l such that a subsequence of $\mathcal{P}_{ij}$ (N_i,N_{m1},N_{m2},....,N_j) exists where consecutive nodes in the subsequence share exactly two keys. This ensures that two neighboring nodes i and j can securely use this path to establish a shared key. Our scheme requires nodes to have two keys in common to ensure better security and making it more resilient to node capture.
- Key establishment phase in which neighboring nodes use the paths computed in the key discovery phase to establish new pair-wise shared keys which are used for secure communication.

We introduce a novel key pre-distribution method in which each sensor node is loaded with a set of keys chosen using our proposed sub-grid scheme, which is an extension of the simple scheme described below.

A. Grid Key Vector Assignment

Figure 1 illustrates our simple pre-distribution scheme. First we construct a $\sqrt{n} \times \sqrt{n}$ grid G with n keys such that exactly one key K_{ij} is at each position of the grid. A node N_{ij} gets the keys in row i and column j. Hence each node in this scheme gets a key vector of size $2 \times \sqrt{n} - 1$. Note that in this arrangement every pair of nodes share atleast two keys in common. This ensures that every neighboring node can establish a common shared pair-wise key after the nodes are deployed. Figure 1a shows that sensor node A shares two keys with node B. Although the basic scheme guarantees connectivity, it is not suitable for sensor networks because it requires a significant number of keys to be allocated to each node. The proposed sub-grid based scheme trades off direct connectivity for key space in the nodes by reducing the key vector size in each node.

B. Sub-Grid Key Vector Assignment

This scheme is an extension of the simple *key vector* assignment explained above. In this scheme the keys are placed in a grid G of size $\sqrt{n} \times \sqrt{n}$ which is divided into $k \times k$ cells each consisting of $m \times m (m = \sqrt{n}/k)$ keys as shown in Fig. 2. A node in a particular cell is assigned the keys from the sub grid as explained below.

Notations

$$= \begin{cases} K_{ij} & \text{A unique key placed at position (ij) on the grid} \\ N_{ij} & \text{Node at position (ij) on the grid} \\ C_{xy} & \text{Represents a cell in the grid } G \\ SG_{xy} & \text{The grid formed by the cell } C_{xy} \text{ and its 8 adjacent cells} \\ V_{ij} & \text{The key vector for a node } N_{ij} \text{ in } SG_{xy} \end{cases}$$

We define a sub-grid SG_{xy} for a cell C_{xy}, which includes the cell itself and all its adjacent cells. For cells at the boundaries, adjacent cells also include cells at the respective opposite boundaries (wraparound). A node N_{ij} in cell C_{xy} is assigned keys from SG_{xy}. The key vector for the node is

$$V_{ij} = \bigcup K_{i,c} + \bigcup K_{r,j},$$

where
$(y-1)\ mod\ k \times m < c < (y+1)\ mod\ k \times m$ and
$(x-1)\ mod\ k \times m < r < (x+1)\ mod\ k \times m$.

The size of the key vector V_{ij} is $6 \times \sqrt{n}/k - 1$. This is considerably smaller than the number of keys required in grid key vector scheme. The number of keys shared by two nodes N_{i1j1} and N_{i2j2}

$$= \begin{cases} 3 \times \frac{\sqrt{n}}{k} & N_{i1j1},\ N_{i2j2} \text{ are in the same cell and } i1 = i2 \text{ or } j1 = j2 \\ 2 \times \frac{\sqrt{n}}{k} & N_{i1j1},\ N_{i2j2} \text{ are in cells which have common sides and } i1 = i2 \text{ or } j1 = j2 \\ 2 & \text{if i1} \neq \text{i2 and j1} \neq \text{j2 and } N_{i1j1} \text{ and } N_{i2j2} \text{ are adjacent} \\ 0 & Otherwise \end{cases}$$

The key vector size depends on the parameter k which determines the number of keys shared between nodes. A higher value of k reduces the key vector size at a node thereby decreasing the memory requirements for the keys. Also, a lower value of k decreases the security of the scheme as capture of a single node discloses a large number of keys. However, a very large value of k will produce lesser sharing of keys among nodes thereby decreasing the connectivity of the network. A suitable value of k should be chosen to maximize connectivity while satisfying stringent memory constraints of a sensor node.

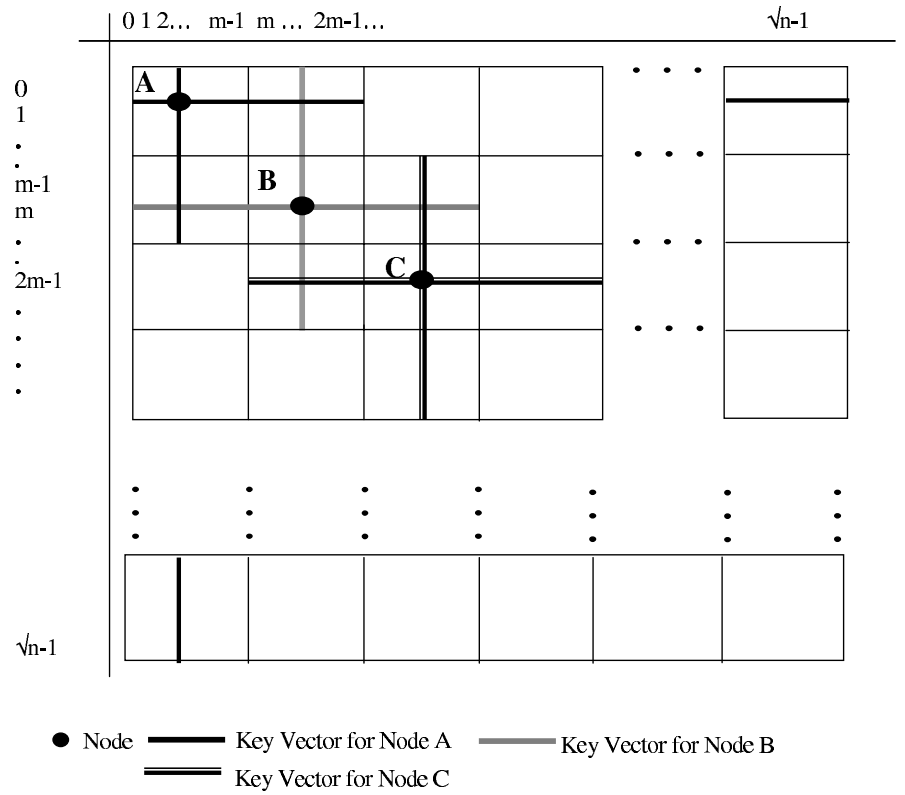


Fig. 2. Sub grid scheme key vector assignment

Our scheme can be extended to use polynomials instead of keys as shown in our earlier work [10]. The extended polynomial scheme uses just one polynomial for each row and column in the grid. Each polynomial is divide into $\sqrt{n}$ shares and placed at the $\sqrt{n}$ positions in the column or row. In each mapping the node gets all the polynomial shares in the column and row of the subgrid. Using the polynomial shares the physically adjacent nodes can construct common shared keys.

C. Loaction awareness and neighbor discovery

To implement the algorthm no loaction information is necessary. A node only tries to estbalish pair wise shared keys only with nodes within comunication range. Neighboring nodes can exchange messages as follows. A node encrypts a message with its avialable keys individually and broadcasts the encrypted messages. All those nodes which have atleast one common key with the sender can reply by decrpyting using that key. Using this approach neighboring nodes can exchange and discover common keys without requiring location information.

D. Key Discovery Phase

Nodes loaded with their key vectors are randomly deployed in the area of interest. After the node deployment phase neighboring nodes exchange their node-id's to determine the number of keys they share. This can be done as the node-id can be used to determine the cell of the node that identifies the *key vector* of the node. Only neighboring nodes that share exactly two keys are allowed to securely communicate with each other by establishing a common shared key to form a direct link. Nodes belonging to the same cell and in the same row or column share more number of keys. However these two nodes are not allowed to use the common keys because capturing of a single node in that row or column reveals those keys.

E. Path-Key Establishment Phase

On completion of the key discovery phase all the neighboring nodes may not have established common shared keys. In order that a node establishes keys with non-key-neighbors, it must go through the path-key establishment phase. In this phase, a node searches among its key-neighbors recursively to find a key-path to the non-key-neighbor. For example in Fig. 2 node A and node C are non-key-neighbors. In order for node A to communicate with node C it must find a intermediate node such that it shares keys with nodes A and C.

III. Security Analysis

Nodes deployed in hostile environments are prone to capture. Capture of a single node discloses all the information about the keys contained in them. An adversary can capture multiple nodes and use these keys to eavesdrop upon links. Hence the security of these keys is very essential for the overall security of the protocol.

We make the following assumptions about the adversary

- We assume that an adversary can capture only a fixed number of nodes in the network.

- Once a node is captured all the information is known to the adversary.
- The adversary can eavesdrop on any link and can decrypt the messages using the known key pairs.

A. Link Capture

We define *link capture* as the ability of the adversary to eavesdrop on the links between any two nodes. The adversary can decrypt the messages on a particular link, if the captured nodes disclose the keys shared between the nodes forming the link. This is a reasonable assumption because for x keys discovered it has to only check for $x \times (x-1)$ combinations.

We use the following metrics to analyse our protocol:

(a) N_c: Number of nodes to be captured to compromise a single link across two specified nodes.
(b) $P(x)$ probability of link compromise for x captured nodes.
(c) $N(x)$ Number of links compromised for x captured nodes.

(a) N_c

We calculate the expected number of nodes to be compromised for capturing a particular link. First we calculate the probability that x nodes are captured for single link compromise. Then we find the expectation over all values of x.

Let $N_{capt}(x)$ be the number of ways of capturing x nodes such that the two keys are known.

$M(x)$ be the number of ways of capturing x nodes such that the two keys are not known:

$$N_{capt}(x) = \binom{n}{y} - M(x)$$

where

$$\begin{aligned} M(x) &= \text{None of the two keys captured +} \\ &\quad \text{Only one key is captured} \\ &= \binom{n-12m+4}{x} + 2 \times \binom{n-6m+1}{x} - \\ &\quad 2 \times \binom{n-12m+4}{x} \end{aligned}$$

$N_c = \frac{\sum x \times N_{capt}(x)}{\sum N_{capt}(x)}$, where $2 \le x \le n$

(b) $P(x)$

We calculate the probability of link compromise for x captured nodes:

$P(x)$=1- $P^{'}(x)$ where $P^{'}(x)$ is probability of link not compromised for x captured nodes.

$$\begin{aligned} P^{'}(x) &= \text{Probability that the two keys are not known + Probability that only one key is known.} \\ &= \frac{\binom{n-12m+4}{x} + 2 \times \binom{n-6m+1}{x} - 2 \times \binom{n-12m+4}{x}}{\binom{n}{x}} \end{aligned}$$

(c) $N(x)$

Let Y be random variable which reprints the number of compromised links, when x nodes are captured.

By using binomial distribution for probabilities we get $Prob(Y=r) = \binom{n}{r} P(x)^r (1-P(x))^{n-r}$

The expected value of the number of captured links, $N(x) = E(Y) = \sum r.Prob(Y=r)$

Theorem

The expected value of the number of links captured for x compromised nodes $E(Y)$ is $\mathcal{L}P(x)$, where $\mathcal{L}$ is the number of links between the deployed nodes.

Proof

Let $Q(x) = 1 - P(x)$

We prove the theorem by using the binomial expansion

$$\begin{aligned} (P(x)+Q(x))^{\mathcal{L}} &= \sum \binom{y}{r} P(x)^r Q(x)^{\mathcal{L}-r} \\ &\quad \text{Differentiating W.R.T } P(x) \text{ on both sides we get the below equation.} \\ 0 &= \sum r\binom{y}{r} P(x)^{r-1} Q(x)^{\mathcal{L}-r} - \\ &\quad \sum (\mathcal{L}-r)\binom{y}{r} P(x)^r Q(x)^{\mathcal{L}-r-1} \\ &\quad \text{Multiplying both sides by } P(x)Q(x) \text{ the expression is simplified to.} \\ 0 &= Q(x)E(Y) + P(x)E(Y) - \\ &\quad \mathcal{L}P(x) \\ \text{Hence, } E(Y) &= \mathcal{L}P(x) \end{aligned}$$

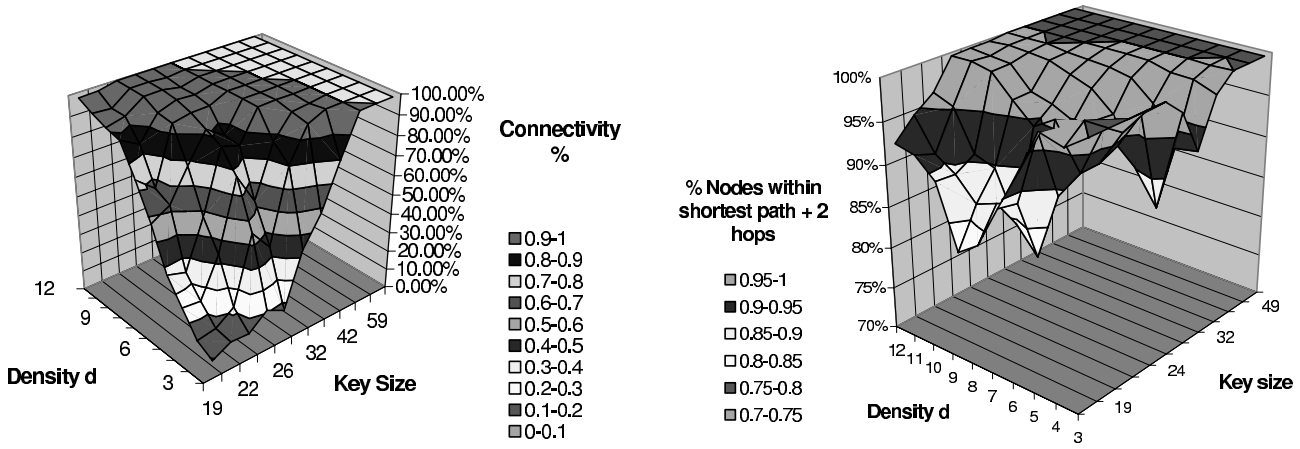
A smaller value of k here increases the connectivity of the network. However the security of the scheme is compromised by making the value of k too small. The value of k can be tuned to provide the desired balance between security and connectivity. Typically, higher connectivity of a network trades off different security issues. In the proposed sub grid key vector key pre-distribution scheme the tradeoff depends on the parameter k. The higher the value of k the smaller is the size of each cell. Consequently, the connectivity is reduced and the security is better.

IV. Simulation

The effectiveness of the sub-grid key vector key pre-distribution scheme is tested through simulation. In the reminder of the section we show the impact of the value of k on connectivity under our subgrid scheme. Next we show the memory savings in our scheme compared to the random key pre-distribution scheme. Connectivity and average path length metrics were calculated for varying values of density, *key vector* size and network size.

A. Experimental Setup

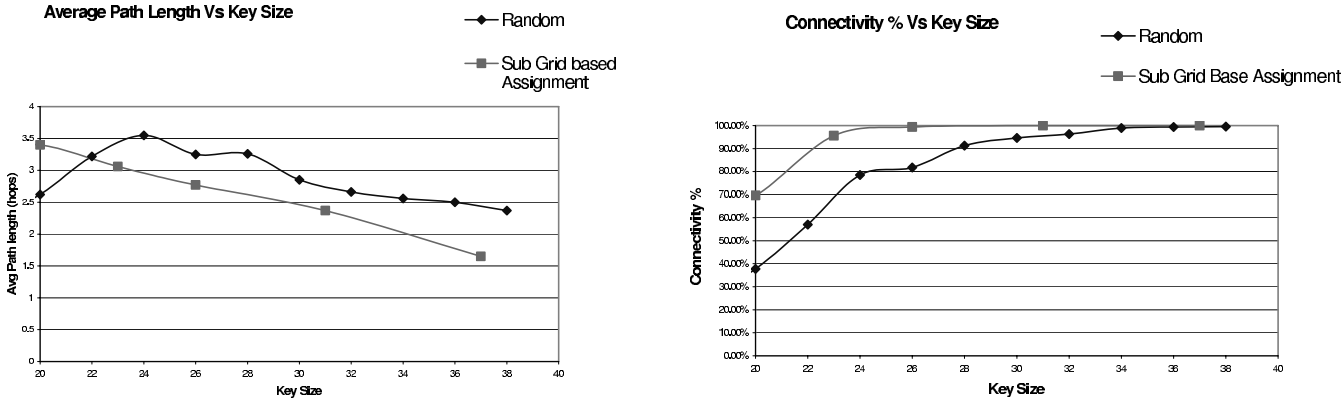
The simulation assumes that nodes are deployed randomly in the target region. The deployment is done with varying densities d. We assumed a two hop neighborhood for our



(a) Connectivity versus Key Vector Size versus Node Density

(b) Percentage of Nodes Within sp+2 versus Key Vector Size versus Node Density

Fig. 3. Impact of Key Vector Size and Node Density On Connectivity and Path Length



(a) Average Path Length versus Key Vector Size

(b) Network Connectivity versus Key Vector Size

Fig. 4. Performance Comparison between Random Scheme and Our Proposed Sub-Grid Scheme

simulations. For our simulations a density d is equivalent to $4\pi d$ nodes in a cluster. The simulations are done for different values of the sub-grid parameter k and density d. d determines the average number of nodes that lie in the neighborhood of a node. Each simulation was run 100 times with different seeds for the random number generator for deployment of nodes and the results presented are the average of 100 runs.

The different phases of the key pre-distribution scheme have been simulated. The logical key space is first defined in the form of a grid and the nodes are distributed keys depending on their position in the grid that determines their identity based on the sub-grid key vector scheme described above. The nodes are then deployed at random. In the next phase a node finds out the nodes within its neighborhood it shares keys with. The cell to which a node belongs can be determined based on the identity of the node. The nodes share keys as specified by the key pre-distribution scheme in Section 2. Next we determine the connectivity of the network. With respect to a particular node, we determine if a path can be established between that node and every other node in the neighborhood, thus determining the connectivity of the network.

B. Impact of Key Vector Size *on Connectivity and Path lengths*

Figure 3a illustrates the relationship between *key vector* size, density and connectivity for a 2500 node network. The graph gives a clear picture of the key vector size that needs to be selected to achieve the desired connectivity for a given density of node distribution. We can observe that the key vector of 42 keys gives more than 98% connectivity for a very sparse neighborhood consisting of 37 nodes. The optimal *key vector* size and density for more than 98% connectivity is along the edge of the cliff in the 3D-graph (Fig. 2a). Note that as density increases the number of keys required to achieve desired 98% connectivity decreases. Figure 3b shows percentage of nodes within $sp+2$(shortest path distance + 2 according to the actual deployment) in relation to *key vector* size and density. Larger

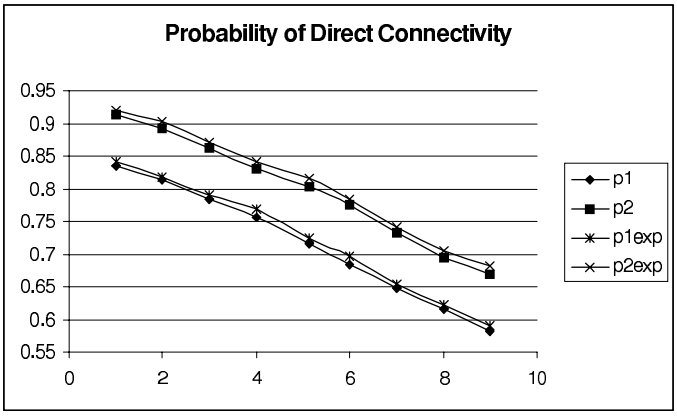


Fig. 5. *Probability That two Nodes Share a Key with increasing value of (k,M) where k=M*

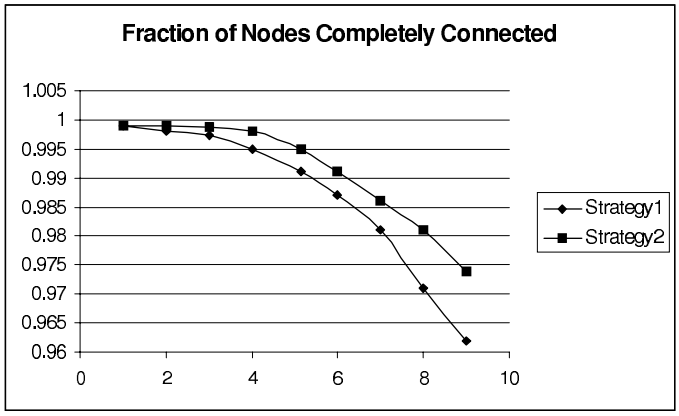


Fig. 6. Fraction of Total Nodes that are Connected with increasing value of (k, M) where $k = M$

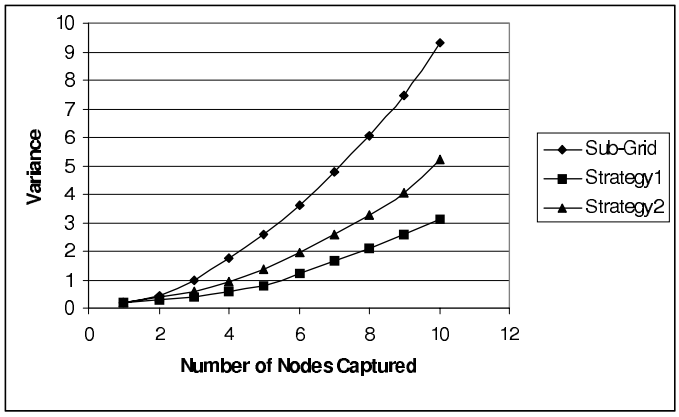


Fig. 7. Variance in Number of Keys Disclosed per Node.

key vector and higher densities increase percentage of nodes within $sp + 2$ increases.

C. Comparison to the Random Scheme

The performance of our protocol is compared with the random key pre-distribution scheme [3]. The number of nodes n that are used in the predeployment was fixed as 1000 under both the protocols. However we consider a cluster of 60 nodes for our simulation of post deployment performance. We test our protocol with varying values of k. The key pool size for the random scheme is of size 10000 as in [3]. The key vector size under each protocol is same. Figure 4a shows a comparison of the average path length(for establishing shared keys between neighbors) under the two protocols using different key vector sizes. Figure 4b shows the relationship between connectivity and key vector size under the two protocols. Our protocol has better average path length than random scheme as key vector size increases. Although the random scheme is initially better it can be used because at such low key vector sizes its connectivity is very low as shown in Fig. 3b. Our protocol achieves the desired connectivity as low as 40% lesser memory compared to the random scheme.

In the next section we propose two deployment strategies in where each node gets its keys from multiple mappings of nodes to keys.

V. Multiple Layer Pre-Deployment Strategies

Consider M one-to-one mappings of N sensor nodes to random positions in the grid G. Our preliminary results indicate that a choice of a small constant for M yields good security-performance trade-offs. In the given mapping i, $1 \leq i \leq M$, each node is assigned a set of sub-row and sub-column keys from its subgrid and adjacent grids as shown in Fig. 2. For a given network deployment ,two physically adjacent sensors can encrypt messages using shared keys under one or more of these mappings.

- *Strategy 1* First keys are placed at each position in the grid. Then for a mapping the nodes are placed at each position on the grid randomly. The keys are assigned to nodes as in the sub-grid scheme. We impose a restriction on the placement of nodes in each mapping. The mappings are such that a node does not get duplicate keys from different mappings. So, every node gets a unique set of keys in each mapping.
- *Strategy 2* Unlike the previous strategy initially each grid position is occupied by a node. Subsequently in each mapping keys are placed randomly at grid positions. Similar to *strategy 1*, each node gets unique set of keys in each mapping

A. Key-Node Mapping Algorithm

Now we present the algorithm to assign the keys to nodes in each mapping i,$1 \leq i \leq M$. Let $pos[j]$ be the array of sets representing the available positions for nodes/keys at the beginning of each mapping. Initially $pos[j]$ contains all the points in the grid every node/key. The following steps are done for each mapping i:

- Each node/key j,$1 \leq j \leq n$, is mapped to a position in the grid such that it can be placed in grid positions available to j from $pos[j]$.
- For each node/key j $pos[j]$ is updated to the new set of available positions. This new set is obtained by removing the newly assigned keys/nodes to node/key j from the set $pos[j]$.

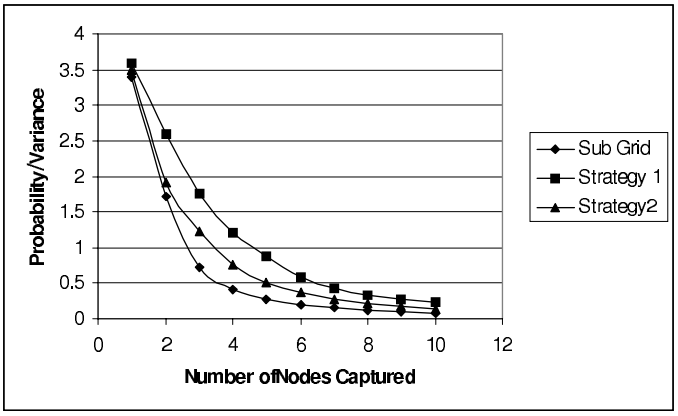


Fig. 8. Probability of Connectivity / Variance of Keys Disclosed

Note that the above approach is used for both the strategies presented above. Nodes and keys can be interchanged in the above algorithm to obtain mappings for the above two strategies. In the next section we evaluate the connectivity and security for the above two schemes.

VI. Connectivity

After the nodes deployed randomly in the area of deployment nodes within communication range try to establish pair-wise keys if they have common shared keys. Hence the probability that a given pair of nodes share at least one key, is a good metric for evaluating the connectivity.

(a) Probability of sharing a key p_1 under *strategy 1* .
The placement of the nodes in each mapping is not completely independent of previous mapping. The following equation gives the probability p that two nodes a share a key.

$$p_1 = \text{Probability that two nodes share a key}$$

$$p_1 = \text{1 - Probability that two nodes do not share a key in any of the mappings}$$

$$p_1 = 1 - \frac{(n-a)(n-b-a)(n-2b-a)\ldots(n-Mb-a)}{n(n-b)(n-2b)\ldots(n-Mb)}$$

$$\text{where} a = \frac{9n}{k^2} + \frac{4\sqrt{n}}{k}, b = \frac{6\sqrt{n}}{k} - 1$$

(b) Probability of sharing a key p_2 under *strategy 2* .
Under this strategy two nodes share keys in every mapping if they are in adjacent cells. Otherwise they share a key if the same key is assigned to the nodes in different mappings.

$$p_2 = \text{Probability that two nodes share a key}$$

$$p_2 = \text{Probability that two nodes share a key in every mapping + Probability that two nodes are not adjacent and share a key}$$

$$p_2 = \frac{10\sqrt{n}}{k} + (1 - \frac{10\sqrt{n}}{n-1})(\frac{(n-2b)(n-3b)\ldots(n-Mb)}{(n-b)(n-2b)\ldots(n-(M-1)b)})$$

Figures 5 and 6 show the analytical and experimental connectivity results for the above two proposed schemes. 10000 nodes were considered for this example. The values for k ans M are chosen such that the number of keys each node gets remains the same. The values for p_1 ,p_2, p_1^{exp} and p_2^{exp} are calculated for different values of k and M. Also the overall connectivity of the network is obtained through simulations. This is the percent of the nodes that are connected after the completion of the key discovery and key path establishment phase.

The values p_1^{exp} and p_2^{exp} are values obtained through simulations. Although direct connectivity comes down with increase in value of M and k overall total connectivity comes down at a much slower rate.

VII. Security Analysis

Nodes are grouped together under strategy2 and then keys are assigned to nodes. However in strategy1 keys are grouped together and then nodes are assigned keys. Therefore under strategy2 the capture of a single node discloses a number of keys which are shared by many nodes. But the capture of nodes under strategy1 does not disclose a significant portion of the keys of all the nodes. The captured keys are distributed evenly among the whole nodes in the network. However under strategy2 a large number of keys for some of the node's are disclosed. So, the whole set of disclosed keys have to be revoked and the number of available keys in some of the nodes is reduced significantly. But while using strategy1 individual nodes still have a large number of available keys. Although strategy1 gives lower connectivity it is better suited owing to its better security performance.

Figures 7 and 8 show the security performance of the protocol. Obviously strategy1 has the lowest variance in the number of keys revealed per node. Under strategy2 and subgrid schemes the position of the nodes is fixed the keys are placed on them. Hence capture a of a single node reveals keys of all its neighbors. Similarly, the connectivity security shows the better performance of strategy1.

VIII. Conclusion

This paper presents a new pre-distribution scheme for wireless networks. It has nearly 40% lesser memory requirements compared to the random scheme. Our detailed simulation shows the gain in memory savings and better connectivity under our scheme compared to the random scheme. This makes our scheme more scalable compared to the previous schemes.

References

[1] R. Kalidindi, V. Parachuri, S. Basavaraju, C. Mallanda, A. Kulshrestha, L. Ray, R. Kannan and A. Durresi. Sub-Grid Based Key Vector Assignment: A Key Pre-Distribution Scheme For Distributed Sensor Networks. *ICWN 2004*.

[2] S. Shakkottai, R. Srikant and N. Shroff. Unreliable Sensor Grids: Coverage, Connectivity and Diameter. *Proc. of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Infocom2003)*, San Francisco, CA, April 2003.

[3] L. Eschenauer and V. D. Gligor. A key management scheme for distributed sensor networks. *Proceedings of the 9th ACM Conference on Computer and Communication Security*, November 2002, pp. 41–47.

[4] H. Chan, A. Perrig and D. Song. Random Key Predistribution Schemes for Sensor Networks. *Proc. of the IEEE Security and Privacy Symposim 2003*, May 2003.

[5] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. Tygar. SPINS: Security Protocols for Sensor Networks. *Proc. of Seventh Annual ACM International Conference on Mobile Computing and Networks (Mobicom 2001)*, Rome, Italy, July 2001.

[6] D. Liu and P. Ning. Establishing Pairwise Keys in Distributed Sensor Networks. *10th ACM Conference on Computer and Communications Security (CCS '03)*, Washington DC, October 2003.

[7] C. Karlof and D. Wagner. Secure Routing in Sensor Networks: Attacks and Countermeasures. *Proc. of First IEEE Workshop on Sensor Network Protocols and Applications*, May 2003.

[8] R. L. Rivest. The RC5 encryption algorithm. *Workshop on Fast Software Encryption*, 1995, pp. 86–96.

[9] R. L. Rivest, A. Shamir and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120–126, 1978.

[10] R. Kalidindi and R. Kannan. Polynomial based Pre-key distribution schemes for sensor networks. LSU Tech. Rep. LSU-CSC-TR-04-003.